

Abstract

ChessPiece - (Entity)

- Stores, retrieves and can set the current or new position of the piece
- Stores, retrieves and can set the colour of the piece
- Stores chess piece letter identifier

- King
- Queen
- Rook
- Knight
- Pawn
- Bishop

King - (Entity)

- Moves one square in every direction
- Checks and checkmate
- Castling with the Rook

- GameState

CheckKingMove - (Use Case)

- Checks to see if the King is moved correctly.

- King
- GameState

Queen - (Entity)

- Moves multiple squares in every direction

- GameState

checkQueenMove - (Use Case)

- | | |
|---|---|
| <ul style="list-style-type: none">• Checks to see if the Queen is moved correctly | <ul style="list-style-type: none">• Queen• GameState |
|---|---|

Rook - (Entity)

- | | |
|--|---|
| <ul style="list-style-type: none">• Moves multiple squares horizontally and vertically in any direction• Castling with the King | <ul style="list-style-type: none">• GameState |
|--|---|

CheckRookMove - (Use Case)

- | | |
|--|--|
| <ul style="list-style-type: none">• Checks to see if the Rook is moved correctly | <ul style="list-style-type: none">• Rook• GameState |
|--|--|

Knight - (Entity)

- | | |
|---|---|
| <ul style="list-style-type: none">• Moves & attacks in a L shape (2 squares horizontally or vertically, then 1 square in a perpendicular direction) | <ul style="list-style-type: none">• GameState |
|---|---|

CheckKnightMove - (Use Case)

- | | |
|--|--|
| <ul style="list-style-type: none">• Checks to see if the Knight is moved correctly | <ul style="list-style-type: none">• Knight• GameState |
|--|--|

Bishop - (Entity)

- | | |
|---|---|
| <ul style="list-style-type: none">• Moves diagonally in any direction | <ul style="list-style-type: none">• GameState |
|---|---|

CheckBishopMove - (Use Case)

- | | |
|--|--|
| <ul style="list-style-type: none">• Checks to see if the Bishop is moved correctly | <ul style="list-style-type: none">• Bishop• GameState |
|--|--|

Pawn - (Entity)

- | | |
|--|---|
| <ul style="list-style-type: none">• If it's the first move, the pawn can move two squares forward. Otherwise, it can only move one square forward• Moves one square forwards and can only attack diagonally• Convert to either a Queen, Bishop, Rook or Knight if the pawn reaches the other side of the board (opponent side) | <ul style="list-style-type: none">• GameState |
|--|---|

CheckPawnMove - (Use Case)

- | | |
|--|--|
| <ul style="list-style-type: none">• Checks to see if the Pawn is moved correctly | <ul style="list-style-type: none">• Pawn• GameState |
|--|--|

Abstract

CheckPlayerMove - (Use Case)

- | | |
|---|--|
| <ul style="list-style-type: none">• Checks to see if player makes a valid move• If move is valid, GameState accepts the move and updates the board• If move is not valid, returns false | <ul style="list-style-type: none">• Rook• King• Queen• Bishop• Knight• Pawn• GameState |
|---|--|

GameState - (Controller)

- | | |
|--|--|
| <ul style="list-style-type: none">• Keeps track of the board state• Methods for checking whether the game is over (whether there is a checkmate)• Methods for moving each chess piece• Methods for capturing opponent player chess pieces | <ul style="list-style-type: none">• Rook• King• Queen• Bishop• Knight• Pawn• GUI• CheckPlayerMove |
|--|--|

LoadGame - (Controller)

- | | |
|--|--|
| <ul style="list-style-type: none">• Initializes the game• Methods to initialize both player objects/instances | <ul style="list-style-type: none">• Rook• King• Queen• Bishop• Knight• Pawn• Gamestate |
|--|--|

CLIBoard - (Interface)

- | | |
|---|---|
| <ul style="list-style-type: none">• Outputs the current state of the board• Methods to print the game board after each valid player move | <ul style="list-style-type: none">• GameState |
|---|---|